Strict Locality In Morphological Derivations

Alëna Aksënova and Aniello De Santo

Stony Brook University

CLS 53

The University of Chicago
May 26, 2017
Motivations

Our goal is to give arguments towards derivational representations in morphology.

Idea

We can exploit Formal Language Theory to
- abstract from narrow, framework-specific details;
- quantify theoretical intuitions;
- \textit{bonus}: cross-domain complexity parallels.

Spoilers:
- long-distance dependencies can be viewed as local
- descriptions of the patterns are much more succinct
- this results in a reduced computational complexity
Motivations

Our goal is to give arguments towards derivational representations in morphology.

Idea

We can exploit Formal Language Theory to
- abstract from narrow, framework-specific details;
- quantify theoretical intuitions;
- *bonus*: cross-domain complexity parallels.

Spoilers:
- long-distance dependencies can be viewed as local
- descriptions of the patterns are much more succinct
- this results in a reduced computational complexity
Motivations

Our goal is to give arguments towards derivational representations in morphology.

Idea

We can exploit Formal Language Theory to

- abstract from narrow, framework-specific details;
- quantify theoretical intuitions;
- \textit{bonus}: cross-domain complexity parallels.

Spoilers:

- long-distance dependencies can be viewed as local
- descriptions of the patterns are much more succinct
- this results in a reduced computational complexity
Outline

1. Derived vs. Derivational
2. Subregular Morphology
3. Russian Nominalization
4. SL Derivations
5. Conclusion
Morphological representations

How to evaluate the grammaticality of morphological forms?

- **Derived sequences**: evaluating the resulting sequence after all operations were applied.
  ⇒ Under this perspective, morphology is not hierarchical, it is simply concatenation of smaller strings.
  (McGregor 2003), i.a.

- **Derivational sequences**: instead of looking at the output, considering the operations that were applied to the root node.
  ⇒ Under this perspective, morphology is hierarchical, and the order in which the affixes were applied matters.
  (Vikner & Vikner 2008), i.a.
Morphological representations

How to evaluate the grammaticality of morphological forms?

- **Derived sequences:** evaluating the resulting sequence after all operations were applied.
  ⇒ Under this perspective, morphology is not hierarchical, it is simply concatenation of smaller strings.
  (McGregor 2003), i.a.

- **Derivational sequences:** instead of looking at the output, considering the operations that were applied to the root node.
  ⇒ Under this perspective, morphology is hierarchical, and the order in which the affixes were applied matters.
  (Vikner & Vikner 2008), i.a.
Morphological representations

How to evaluate the grammaticality of morphological forms?

- **Derived sequences**: evaluating the resulting sequence after all operations were applied.
  ⇒ Under this perspective, morphology is not hierarchical, it is simply concatenation of smaller strings.
  (McGregor 2003), i.a.

- **Derivational sequences**: instead of looking at the output, considering the operations that were applied to the root node.
  ⇒ Under this perspective, morphology is hierarchical, and the order in which the affixes were applied matters.
  (Vikner & Vikner 2008), i.a.
# Derived vs. derivational approaches: an example

<table>
<thead>
<tr>
<th>Derived approach</th>
<th>Derivational approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Semantics is extracted based on the form of the string</td>
<td>1. Semantics is extracted based on the order of applied operations</td>
</tr>
<tr>
<td>2. Representation:</td>
<td>2. Representation:</td>
</tr>
<tr>
<td>un + lock + able</td>
<td>[lock + able] + un [lock + un] + able</td>
</tr>
</tbody>
</table>

**un-lock-able**
Derived vs. derivational approaches: an example

Derived approach

1. Semantics is extracted based on the form of the string
   
2. Representation: 
   \( \text{un} + \text{lock} + \text{able} \)

Derivational approach

1. Semantics is extracted based on the order of applied operations

2. Representation: 
   \([\text{lock} + \text{able}] + \text{un}\) 
   \([\text{lock} + \text{un}] + \text{able}\)
Derived vs. derivational approaches: an example

Derived approach

1. Semantics is extracted based on the form of the string
2. Representation:
   un + lock + able

Derivational approach

1. Semantics is extracted based on the order of applied operations
2. Representation:
   [lock + able] + un
   [lock + un] + able
One topic of debate: Semantic ambiguity

The derivational representation captures the semantic ambiguity caused by different order of affix application.

Can we approach this problem from a more formal point of view?
One topic of debate: Semantic ambiguity

The derivational representation captures the semantic ambiguity caused by different order of affix application.

Can we approach this problem from a more formal point of view?
One topic of debate: Semantic ambiguity

The derivational representation captures the semantic ambiguity caused by different order of affix application.

\[
\text{un-lock-able}
\]

\[
\text{[un-lock]-able} \quad \text{# possible to unlock}
\]
\[
\text{stem-un-able}
\]

\[
\text{un-[lock-able]} \quad \text{# not possible to lock}
\]
\[
\text{stem-able-un}
\]

Can we approach this problem from a more formal point of view?
One topic of debate: Semantic ambiguity

The derivational representation captures the semantic ambiguity caused by different order of affix application.

\[
\text{un-lock-able} \\
\text{[un-lock]-able} \quad \text{# possible to unlock} \\
\text{stem-un-able} \\
\text{un-[lock-able]} \quad \text{# not possible to lock} \\
\text{stem-able-un}
\]

Can we approach this problem from a more formal point of view?
One topic of debate: Semantic ambiguity

The derivational representation captures the semantic ambiguity caused by different order of affix application.

un-lock-able

[un-lock]-able
  # possible to unlock
stem-un-able

un-[lock-able]
  # not possible to lock
stem-able-un

Can we approach this problem from a more formal point of view?
The Chomsky Hierarchy of String Languages

Languages (stringsets) can be classified according to the complexity of the grammars that generate them.

- recursively enumerable
- context-sensitive
- mildly-context sensitive
- context-free
- regular
- (finite)
The Chomsky Hierarchy of String Languages

Languages (stringsets) can be classified according to the complexity of the grammars that generate them.

The hierarchy includes:
- *recursively enumerable*
- *context-sensitive*
- *mildly-context sensitive*
- *context-free*
- *regular*
- *(finite)*

- **Phonology**
  - Kaplan and Kay (1994)
- **Syntax**
  - Shieber (1985)
- **Morphology**
  - Karttunen et al. (1992)
Morphology as a Regular System

precisely predictions for:

- typology → e.g. no unbounded center embedding
- learnability → e.g. no Gold learning for regular languages
- cognition → e.g. finitely bounded working memory
Subregular Hierarchy

Not full power of finite machinery is needed
⇒ subregular hierarchy

- Subregular hierarchy introduced
  (McNaughton&Papert 1971)
- Subregular hierarchy expanded
  (Rogers et. al 2010)
- Phonology is subregular
  (Heinz&Idsardi 2013)
- Morphotactics is subregular
  (Aksenova et. al 2016)

Subregular Morphotactics

Morphotactic dependencies seem to be
- strictly local (SL)
- tier-based strictly local (TSL)
Not full power of finite machinery is needed
⇒ subregular hierarchy

- Subregular hierarchy introduced
  (McNaughton & Papert 1971)
- Subregular hierarchy expanded
  (Rogers et al. 2010)
- Phonology is subregular
  (Heinz & Idsardi 2013)
- Morphotactics is subregular
  (Aksenova et al. 2016)

Subregular Morphotactics

Morphotactic dependencies seem to be
- strictly local (SL)
- tier-based strictly local (TSL)
Not full power of finite machinery is needed ⇒ **subregular hierarchy**

- Subregular hierarchy introduced (McNaughton & Papert 1971)
- Subregular hierarchy expanded (Rogers et al. 2010)
- Phonology is subregular (Heinz & Idsardi 2013)
- Morphotactics is subregular (Aksenova et al. 2016)

**Subregular Morphotactics**

Morphotactic dependencies seem to be

- strictly local (SL)
- tier-based strictly local (TSL)
**Strictly local (SL) grammars** capture *local* dependencies by listing *disallowed substrings.*

**Example (Affixation in English)**

- *un-* is a prefix: un-holy, un-do
- *-able* is a suffix: drink-able, move-able
- \(G_{SL} = \{*\text{able-stem}, *\text{stem-un}\}\)
  - blocks improper ordering
  - predicts co-occurrence of these affixes

Indeed, it is correct:

- \(\text{ok do, ok un-do, ok un-do-able}\)
- *able-move, *able-do-un
SL morphotactics: affixation

Strictly local (SL) grammars capture local dependencies by listing disallowed substrings.

Example (Affixation in English)

- **un**- is a prefix: un-holy, un-do
- **-able** is a suffix: drink-able, move-able

\[ G_{SL} = \{ \text{*able-stem, *stem-un} \}\]
- blocks improper ordering
- predicts co-occurrence of these affixes

Indeed, it is correct:
- \( ok \) do, \( ok \) un-do, \( ok \) un-do-able
- *able-move, *able-do-un
**Strictly local (SL) grammars** capture *local* dependencies by listing *disallowed* substrings.

### Example (Affixation in English)

- **un-** is a prefix: un-holy, un-do
- **-able** is a suffix: drink-able, move-able
- \( G_{SL} = \{ *\text{able-stem}, \ *\text{stem-un} \} \)
  - blocks improper ordering
  - predicts co-occurrence of these affixes

- Indeed, it is correct:
  - \( \textit{ok do}, \textit{ok un-do}, \textit{ok un-do-able} \)
  - \( *\text{able-move}, *\text{able-do-un} \)
Strictly local (SL) grammars capture local dependencies by listing disallowed substrings.

Example (Affixation in English)

- *un-* is a prefix: un-holy, un-do
- *-able* is a suffix: drink-able, move-able
- \( G_{SL} = \{ \text{*able-stem, *stem-un} \} \)
  - blocks improper ordering
  - predicts co-occurrence of these affixes

Indeed, it is correct:

- \( ok \) do, \( ok \) un-do, \( ok \) un-do-able
- *able-move, *able-do-un

The Subregular approach abstracts from theory internal details ...

... while providing a precise notion of complexity of a certain pattern.

Are there complexity differences among different representations (e.g., derived vs. derivational) of the same pattern?
Subregular morphotactics

- The **Subregular approach** abstracts from theory internal details ...
- ... while providing a precise notion of complexity of a certain pattern.

Are there complexity differences among different representations (eg., derived vs. derivational) of the same pattern?
A case study from Russian

**Russian aspectual metamorphosis**
- Stems are intrinsically atelic;
- *telic* prefixes and *atelic* suffix;
- telic prefix can be added only to the atelic form;
- atelic suffix can be added only to the telic form.

**Russian nominalization**
The nominalization suffix:
- cannot apply directly to the stem;
- cannot apply to a telic form;
- can only be applied after the stem is converted to an atelic form.
Russian aspectual metamorphosis

- atelic
  - ok
  - kry-t'
  - open-INF
- telic
  - ok
  - ot-kry-t'
  - TEL-open-INF
- atelic
  - ok
  - ot-kry-va-t'
  - TEL-open-ATEL-INF
- telic
  - ok
  - na-ot-kry-va-t'
  - TEL-TEL-open-ATEL-INF
- atelic
  - *kry-va-t'
  - open-ATEL-INF
- telic
  - *na-ot-kry-t'
  - TEL-TEL-open-INF
- atelic
  - *ot-kry-va-va-t'
  - TEL-open-ATEL-ATEL-INF
Russian aspectual metamorphosis

**Derived vs. Derivational**

**Subregular Morphology**

**Russian Nominalization**

**SL Derivations**

**Conclusion**

- **atelic**
  - *ok* kry-t'
  - open-INF

- **telic**
  - *ok* ot-kry-t'
  - TEL-open-INF

- **atelic**
  - *ok* ot-kry-va-t'
  - TEL-open-ATEL-INF

- **telic**
  - *ok* na-ot-kry-va-t'
  - TEL-TEL-open-ATEL-INF

- **atelic**
  - *ok* ot-kry-va-va-t'
  - TEL-TEL-open-ATEL-ATEL-INF

- **telic**
  - *ok* na-ot-kry-va-t'
  - TEL-TEL-open-ATEL-INF

- **atelic**
  - *ok* ot-kry-va-t'
  - TEL-open-ATEL-INF

- **telic**
  - *ok* ot-kry-t'
  - TEL-open-INF
Russian aspectual metamorphosis

- **Atelic**
  - kry-t'
  - open-INF

- **Telic**
  - ot-kry-t'
  - TEL-open-INF

- **Atelic**
  - *kry-va-t'
  - open-ATEL-INF

- **Telic**
  - na-ot-kry-t'
  - TEL-TEL-open-INF

- **Atelic**
  - ot-kry-va-t'
  - TEL-open-ATEL-INF

- **Telic**
  - na-ot-kry-va-t'
  - TEL-TEL-open-ATEL-INF

- **Atelic**
  - *ot-kry-va-va-t'
  - TEL-open-ATEL-ATEL-INF
Russian aspectual metamorphosis

1. **atelic**

- ok kry-t’
- open-INF

2. **telic**

- ok ot-kry-t’
- TEL-open-INF

3. **telic**

- *na-ot-kry-t’
- TEL-TEL-open-INF

4. **telic**

- ok na-ot-kry-va-t’
- TEL-TEL-open-ATEL-INF

5. **atelic**

- ok ot-kry-va-t’
- TEL-open-ATEL-INF

6. **atelic**

- *kry-va-t’
- open-ATEL-INF

7. **atelic**

- *ot-kry-va-va-t’
- TEL-open-ATEL-ATEL-INF
A case study from Russian

**Russian aspectual metamorphosis**

- Stems are intrinsically atelic;
- *telic* prefixes and *atelic* suffixes;
- telic prefix can be added only to the atelic form;
- atelic suffix can be added only to the telic form.

**Russian nominalization**

The nominalization suffix:
- cannot apply directly to the stem;
- cannot apply to a telic form;
- can only be applied after the stem is converted to an atelic form.
Russian -nie nominalization

- **Atelic**
  - `kry-t'` (open-INF)
  - `ok` + `ot-`:
    - `telic` `ok` `ot-kry-t'` (TEL-open-INF)
  - `ok` + `-va`:
    - `telic` `ok` `ot-kry-va-t'` (TEL-open-ATEL-INF)
  - `ok` + `na-`:
    - `telic` `ok` `na-ot-kry-va-t'` (TEL-TEL-open-ATEL-INF)

- **Telic**
  - `ot-kry-t'` (TEL-open-INF)
  - `ok` + `ot-`:
    - `*ot-kry-nie` (TEL-open-NMN)
  - `ok` + `-va`:
    - `ot-kry-va-nie` (TEL-open-ATEL-NMN)
  - `ok` + `na-`:
    - `na-ot-kry-va-nie` (TEL-TEL-open-ATEL-NMN)
Russian -nie nominalization

**atelic**
- kry-t’
- open-INF

**telic**
- ot-kry-t’
- TEL-open-INF

Examples:
- *kry-nie* open-NMN
- *ot-kry-nie* TEL-open-NMN
- *na-ot-kry-va-nie* TEL-TEL-open-ATEL-NMN
Russian -nie nominalization

<table>
<thead>
<tr>
<th>Derived vs. Derivational</th>
<th>Subregular Morphology</th>
<th>Russian Nominalization</th>
<th>SL Derivations</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

**atelic**
- **kry-t’**
- **open-INF**

**telic**
- **ot-kry-t’**
- **TEL-open-INF**

**atelic**
- **ot-kry-va-t’**
- **TEL-open-ATEL-INF**

**telic**
- **na-ot-kry-va-t’**
- **TEL-TEL-open-ATEL-INF**

**atelic**
- **ot-kry-va-t’**
- **TEL-open-ATEL-INF**

**telic**
- **na-ot-kry-va-t’**
- **TEL-TEL-open-ATEL-INF**

**atelic**
- **ok**
- **kry-nie**
- **open-NMN**

**telic**
- **ok**
- **ot-kry-nie**
- **TEL-open-NMN**

**atelic**
- **ok**
- **ot-kry-va-nie**
- **TEL-open-ATEL-NMN**

**telic**
- **ok**
- **na-ot-kry-va-nie**
- **TEL-TEL-open-ATEL-NMN**

**atelic**
- **ok**
- **kry-nie**
- **open-NMN**

**telic**
- **ok**
- **ot-kry-nie**
- **TEL-open-NMN**

**atelic**
- **ok**
- **ot-kry-va-nie**
- **TEL-open-ATEL-NMN**

**telic**
- **ok**
- **na-ot-kry-va-nie**
- **TEL-TEL-open-ATEL-NMN**

14
Russian -nie nominalization

- **Atelic**
  - kry-t'
  - open-INF
  - + ot-
    - **Telic**
      - ot-kry-t'
      - TEL-open-INF
      - + va
        - **Atelic**
          - ot-kry-va-t'
          - TEL-open-ATEL-INF
          - + na-
            - **Telic**
              - na-ot-kry-va-t'
              - TEL-TEL-open-ATEL-INF

- *kry-nie*
  - open-NMN

- *ot-kry-nie*
  - TEL-open-NMN

- *na-ot-kry-va-nie*
  - TEL-TEL-open-ATEL-NMN
A case study from Russian: summary

**Russian aspectual metamorphosis**
- telic prefix can be added only to the atelic form;
- atelic suffix can be added only to the telic form.

**Russian nominalization**
- The nominalization suffix can only be applied after the stem is converted to an atelic form.
### Russian aspectual metamorphosis: a SL account?

<table>
<thead>
<tr>
<th>atelic</th>
<th>telic</th>
<th>atelic</th>
<th>telic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ok kry-t’</td>
<td>ok ot-kry-t’</td>
<td>ok kry-va-t’</td>
<td>ok na-ot-kry-t’</td>
</tr>
<tr>
<td>open-INF</td>
<td>TEL-open-INF</td>
<td>TEL-open-ATEL-INF</td>
<td>TEL-TEL-open-INF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>atelic</th>
<th>telic</th>
<th>atelic</th>
<th>telic</th>
</tr>
</thead>
<tbody>
<tr>
<td>*kry-va-t’</td>
<td>*na-ot-kry-t’</td>
<td>*ot-kry-va-va-t’</td>
<td></td>
</tr>
<tr>
<td>open-ATEL-INF</td>
<td>TEL-TEL-open-INF</td>
<td>TEL-open-ATEL-ATEL-INF</td>
<td></td>
</tr>
</tbody>
</table>
### Russian aspectual metamorphosis: a SL account?

|        | |        | |        | |        |
|--------|--------|--------|--------|--------|--------|
| **atelic** | **ok** | **kry-t’** | **open-INF** | |  |
| **telic** | **ok** | **ot-kry-t’** | **TEL-open-INF** | |  |
| **atelic** | **ok** | **ot-kry-va-t’** | **TEL-open-ATEL-INF** | |  |
| **telic** | **ok** | **na-ot-kry-va-t’** | **TEL-TEL-open-ATEL-INF** | |  |
| **atelic** | **ok** | **kry-va-t’** | **open-ATEL-INF** | |  |
| **telic** | **ok** | **na-ot-kry-t’** | **TEL-TEL-open-INF** | |  |
| **telic** | **ok** | **na-ot-kry-va-t’** | **TEL-open-ATEL-ATEL-INF** | |  |
| **telic** | **ok** | **ot-kry-va-va-t’** | **TEL-open-ATEL-ATEL-INF** | |  |
Russian aspectual metamorphosis: a SL account?

**Atelic**

- *ok kry-t’*
  - open-INF

**Telic**

- *ok ot-kry-t’*
  - TEL-open-INF

**Atelic**

- *ok ot-kry-va-t’*
  - TEL-open-ATEL-INF

**Telic**

- *na-ot-kry-t’*
  - TEL-TEL-open-INF

**Atelic**

- *kry-va-t’*
  - open-ATEL-INF

**Telic**

- *na-ot-kry-va-t’*
  - TEL-TEL-open-ATEL-INF

- *ot-kry-va-va-t’*
  - TEL-open-ATEL-ATEL-INF
Russian aspectual metamorphosis: a SL account?

**atelic**

\( ok \) kry-t’

open-INF

**telic**

\( ok \) ot-kry-t’

TEL-open-INF

\[ \begin{array}{c}
\text{atelic} \\
\text{\( ok \) ot-kry-va-t’} \\
\text{TEL-open-ATEL-INF}
\end{array} \]

\[ \begin{array}{c}
\text{telic} \\
\text{\( ok \) na-ot-kry-va-t’} \\
\text{TEL-TEL-open-ATEL-INF}
\end{array} \]

\[ \begin{array}{c}
\text{atelic} \\
\text{\( *kry-va-t’} \\
\text{open-ATEL-INF}
\end{array} \]

\[ \begin{array}{c}
\text{telic} \\
\text{\( *na-ot-kry-t’} \\
\text{TEL-TEL-open-INF}
\end{array} \]

\[ \begin{array}{c}
\text{telic} \\
\text{\( *ot-kry-va-va-t’} \\
\text{TEL-open-ATEL-ATEL-INF}
\end{array} \]
Russian aspectual metamorphosis: a SL account?

**Atelic**
- ok kry-t'
- open-INF

**Telic**
- ok ot-kry-t'
- TEL-open-INF

**Atelic**
- ok ot-kry-va-t'
- TEL-open-ATEL-INF

**Telic**
- ok na-ot-kry-t'
- TEL-TEL-open-ATEL-INF

**Atelic**
- *kry-va-t'
- open-ATEL-INF

**Telic**
- *na-ot-kry-t'
- TEL-TEL-open-ATEL-INF

**Telic**
- *ot-kry-va-va-t'
- TEL-open-ATEL-ATEL-INF
Russian aspectual metamorphosis: a SL account?

- **atelic**
  - $ok$ kry-t'
  - open-INF

- **telic**
  - $ok$ ot-kry-t'
  - TEL-open-INF

- **atelic**
  - $ok$ ot-kry-va-t'
  - TEL-open-ATEL-INF

- **telic**
  - $ok$ na-ot-kry-va-t'
  - TEL-TEL-open-ATEL-INF

- **atelic**
  - *kry-va-t'
  - open-ATEL-INF

- **telic**
  - *na-ot-kry-t'
  - TEL-TEL-open-ATEL-INF

- **telic**
  - *ot-kry-va-va-t'
  - TEL-open-ATEL-ATEL-INF

[Diagram showing the transformation of aspectual forms in Russian.]
Russian aspectual metamorphosis: a SL account?

**atelic**
- \( ok \) kry-t'
- open-INF

**telic**
- \( ok \) ot-kry-t'
- TEL-open-INF

**atelic**
- \( ok \) ot-kry-va-t'
- TEL-open-ATEL-INF

**telic**
- \( ok \) na-ot-kry-va-t'
- TEL-TEL-open-ATEL-INF

**atelic**
- \( * \) kry-va-t'
- open-ATEL-INF

**telic**
- \( * \) na-ot-kry-t'
- TEL-TEL-open-INF

**telic**
- \( * \) ot-kry-va-va-t'
- TEL-open-ATEL-ATEL-INF
## Russian aspectual metamorphosis: a SL account?

<table>
<thead>
<tr>
<th>atelic</th>
<th>telic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ok</strong> kry-t’</td>
<td><strong>ok</strong> ot-kry-t’</td>
</tr>
<tr>
<td>open-INF</td>
<td>TEL-open-INF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>atelic</th>
<th>telic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ok</strong> ot-kry-va-t’</td>
<td><strong>ok</strong> na-ot-kry-t’</td>
</tr>
<tr>
<td>TEL-open-ATEL-INF</td>
<td>TEL-TEL-open-INF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>atelic</th>
<th>telic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ok</strong> ot-kry-va-va-t’</td>
<td><strong>ok</strong> ot-kry-va-t’</td>
</tr>
<tr>
<td>TEL-TEL-open-ATEL-INF</td>
<td>TEL-open-ATEL-INF</td>
</tr>
</tbody>
</table>
Russian aspectual metamorphosis: derivational approach

- open-INF
- stem-inf
- TEL-open-INF
- stem-tel-inf
- TEL-TEL-open-INF
- stem-tel-tel-inf
- TEL-TEL-open-ATEL-INF
- stem-tel-atel-tel-inf
- open-ATEL-INF
- stem-atel-inf
- open-ATEL-ATEL-INF
- stem-tel-atel-atel-inf
- *open-ATEL-ATEL-ATEL-INF
- stem-tel-atel-atel-atel-inf
Russian aspectual metamorphosis: derivational approach
Russian aspectual metamorphosis: derivational approach

- open-inf
  - stem-inf
    - TEL-open-INF
      - stem-tel-inf
        - *open-ATEL-INF
          - stem-atel-inf
            - *TEL-TEL-open-ATEL-INF
              - stem-tel-tel-inf
                - TEL-TEL-open-ATEL-INF
                  - stem-tel-atel-tel-inf
                - TEL-TEL-open-ADEL-ATEL-INF
                  - *TEL-open-ADEL-ATEL-ATEL-INF
                    - stem-tel-atel-atel-tel-inf
Russian aspectual metamorphosis: derivational approach
Russian aspectual pattern: SL derivations

- **stem-atel**: do not add the atelic suffix to the verbal stem;
  
  *kry-va-t‘
  
  \[ \text{stem-atel-inf} \]

- **tel-tel**: ban telic prefix if it was added right before it;
  
  *na-ot-kry-t‘
  
  \[ \text{stem-tel-tel-inf} \]

- **atel-atel**: ban atelic suffix if it was added right before it.
  
  *ot-kry-va-va-t‘
  
  \[ \text{stem-tel-atel-atel-inf} \]
Russian aspectual pattern: SL derivations

- *stem-atel*: do not add the atelic suffix to the verbal stem;
  
  *kry-va-t’*
  
  stem-atel-inf

- *tel-tel*: ban telic prefix if it was added right before it;
  
  *na-ot-kry-t’*
  
  stem-tel-tel-inf

- *atel-atel*: ban atelic suffix if it was added right before it.
  
  *ot-kry-va-va-t’*
  
  stem-tel-atel-atel-inf
Russian aspectual pattern: SL derivations

- **stem-atel**: do not add the atelic suffix to the verbal stem;
  
  *kry-va-t'  
  stem-atel-inf

- **tel-tel**: ban telic prefix if it was added right before it;
  
  *na-ot-kry-t'  
  stem-tel-tel-inf

- **atel-atel**: ban atelic suffix if it was added right before it.
  
  *ot-kry-va-va-t'  
  stem-tel-atel-atel-inf
Derived vs. derivational representations

\[^{ok} \text{na-ot-kry-va-t'}\]
TEL-TEL-open-ATEL-INF
stem-tel-atel-tel-inf

\[^{*} \text{ot-kry-va-va-t'}\]
TEL-open-ATEL-ATEL-INF
stem-tel-atel-atel-inf

\[\text{na-ot-kry-va-t'}\]
\[\text{stem-tel-atel-tel-inf}\]

\[\text{ot-kry-va-va-t'}\]
\[\text{stem-tel-atel-atel-inf}\]

Derived strings: 4-SL

Derivational strings: 2-SL

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.
### Derived vs. Derivational representations

<table>
<thead>
<tr>
<th>Derived strings: 4-SL</th>
<th>Derivational strings: 2-SL</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{na-ot-kry-va-t'} )</td>
<td>( \text{stem-tel-atel-atel-inf} )</td>
</tr>
<tr>
<td>( \text{TEL-TEL-open-ATEL-INF} )</td>
<td>( \text{stem-tel-atel-atel-inf} )</td>
</tr>
<tr>
<td>( \text{ok na-ot-kry-va-t'} )</td>
<td>( \text{TEL-open-ATEL-ATEL-INF} )</td>
</tr>
<tr>
<td>( \text{stem-tel-atel-tel-inf} )</td>
<td>( \text{stem-tel-atel-atel-inf} )</td>
</tr>
</tbody>
</table>

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.
Derived vs. derivational representations

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.
### Derived vs. derivational representations

<table>
<thead>
<tr>
<th>Derived strings: 4-SL</th>
<th>Derivational strings: 2-SL</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{ok} ) \text{ na-ot-kry-va-t' } )</td>
<td>( \text{stem-tel-atel-tel-inf} )</td>
</tr>
<tr>
<td>( \text{TEL-TEL-open-ATEL-INF} )</td>
<td>( \text{stem-tel-atel-atel-inf} )</td>
</tr>
<tr>
<td>\text{na-ot-kry-va-t'}</td>
<td>\text{stem-tel-[atel-tel]-inf}</td>
</tr>
<tr>
<td>\text{ot-kry-va-va-t'}</td>
<td>\text{stem-tel-[atel-atel]-inf}</td>
</tr>
</tbody>
</table>

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.
Derived vs. derivational representations

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.
Derived vs. derivational representations

4-SL is basically memorizing the whole string without capturing the intuition, whereas 2-SL is the succinct representation of the generalization.
Russian -nie nominalization: a SL account?

The nominalization suffix -nie can be added only after the stem was converted to the atelic form.

A SL account?

\[
\begin{align*}
\text{atelic} & \\
\text{ok} & \text{ot-kry-va-t'} \\
\text{TEL-open-ATEL-INF} & \rightarrow \text{ok} \\
\text{ot-kry-va-nie} & \text{TEL-open-ATEL-NMN}
\end{align*}
\]

\[
\begin{align*}
\text{telic} & \\
\text{ok} & \text{na-ot-kry-va-t'} \\
\text{TEL-TEL-open-ATEL-INF} & \rightarrow \text{ok} \\
\text{na-ot-kry-va-nie} & \text{TEL-TEL-open-ATEL-NMN}
\end{align*}
\]

\[
\begin{align*}
\text{ok} & \text{ot-kry-va-nie} \\
\text{TEL-open-ATEL-NMN} & \\
\text{ok} & \text{na-ot-kry-va-nie} \\
\text{TEL-TEL-open-ATEL-NMN}
\end{align*}
\]
The nominalization suffix -nie can be added only after the stem was converted to the atelic form.

**atelic**

\[
\text{ok} \quad \text{ot-kry-va-t'} \\
\text{TEL-open-ATEL-INF}
\]

\[
\text{ok} \quad \text{ot-kry-va-nie} \\
\text{TEL-open-ATEL-NMN}
\]

**telic**

\[
\text{ok} \quad \text{na-ot-kry-va-t'} \\
\text{TEL-TEL-open-ATEL-INF}
\]

\[
* \quad \text{na-ot-kry-va-nie} \\
\text{TEL-TEL-open-ATEL-NMN}
\]

A **SL account**?

\[
\text{ok} \quad \text{ot-kry-va-nie} \\
\text{TEL-open-ATEL-NMN}
\]

\[
* \quad \text{na-ot-kry-va-nie} \\
\text{TEL-TEL-open-ATEL-NMN}
\]
The nominalization suffix -nie can be added only after the stem was converted to the atelic form.

**A SL account?**

ok ot-kry-va-nie
TEL-open-ATEL-NMN

* na-ot-kry-va-nie
TEL-TEL-open-ATEL-NMN
The nominalization suffix -nie can be added only after the stem was converted to the atelic form.

**Atelic**

\[
\text{ok} \quad \text{ot-kry-va-t'}
\]

TEL-open-ATEL-INF

\[
\text{ok} \quad \text{ot-kry-va-nie}
\]

TEL-open-ATEL-NMN

**Telic**

\[
\text{ok} \quad \text{na-ot-kry-va-t'}
\]

TEL-TEL-open-ATEL-INF

\[
* \quad \text{na-ot-kry-va-nie}
\]

TEL-TEL-open-ATEL-NMN

**A SL account?**

\[
\text{ok} \quad \text{ot-kry-va-nie}
\]

TEL-open-ATEL-NMN

\[
* \quad \text{na-ot-kry-va-nie}
\]

TEL-TEL-open-ATEL-NMN
The nominalization suffix *-nie can be added only after the stem was converted to the atelic form.

**Atelic**

\[ \text{ok} \text{ ot-kry-va-t'} \]
\[ \text{TEL-open-ATEL-INF} \]

\[ \text{ok} \text{ ot-kry-va-nie} \]
\[ \text{TEL-open-ATEL-NMN} \]

**Telic**

\[ \text{ok} \text{ na-ot-kry-va-t'} \]
\[ \text{TEL-TEL-open-ATEL-INF} \]

\[ *\text{na-ot-kry-va-nie} \]
\[ \text{TEL-TEL-open-ATEL-NMN} \]

**A SL account?**

\[ \text{ok} \text{ ot-kry-va-nie} \]
\[ \text{TEL-open-ATEL-NMN} \]

\[ *\text{na-ot-kry-va-nie} \]
\[ \text{TEL-TEL-open-ATEL-NMN} \]
The nominalization suffix -nie can be added only after the stem was converted to the atelic form.

**Atelic**

\[ ok \text{ ot-kry-va-t'} \]
TEL-open-ATEL-INF

\[ ok \text{ ot-kry-va-nie} \]
TEL-open-ATEL-NMN

**Telic**

\[ ok \text{ na-ot-kry-va-t'} \]
TEL-TEL-open-ATEL-INF

\[ *\text{na-ot-kry-va-nie} \]
TEL-TEL-open-ATEL-NMN

**A SL account?**

\[ ok \text{ ot-kry-va-nie} \]
TEL-open-ATEL-NMN

\[ *\text{na-ot-kry-va-nie} \]
TEL-TEL-open-ATEL-NMN
Russian -nie nominalization: derivational account

```
open-INF    -> *open-NMN
  stem-inf    -> stem-nmn

TEL-open-INF    -> *TEL-open-NMN
  stem-tel-inf    -> stem-tel-nmn

TEL-open-ATEL-INF    -> TEL-open-ATEL-NMN
  stem-tel-atel-inf    -> stem-tel-atel-nmn

TEL-TEL-open-ATEL-INF    -> *TEL-TEL-open-ATEL-NMN
  stem-tel-atel-tel-inf    -> stem-tel-atel-tel-nmn
```
Russian -nie nominalization: derivational account

open-INF

*open-NMN

stem-inf

stem-nmn

TEL-open-INF

*TEL-open-NMN

stem-tel-inf

stem-tel-nmn

TEL-open-ATEL-INF

TEL-open-ATEL-NMN

stem-tel-atel-inf

stem-tel-atel-nmn

TEL-TEL-open-ATEL-INF

*TEL-TEL-open-ATEL-NMN

stem-tel-atel-tel-inf

stem-tel-atel-tel-nmn
Russian \textit{-nie} nominalization: derivational account

- \texttt{open-INF}\rightarrow\texttt{stem-inf}\rightarrow\texttt{*open-NMN}\rightarrow\texttt{stem-nmn}

- \texttt{TEL-open-INF}\rightarrow\texttt{stem-tel-inf}\rightarrow\texttt{*TEL-open-NMN}\rightarrow\texttt{stem-tel-nmn}

- \texttt{TEL-open-ATEL-INF}\rightarrow\texttt{stem-tel-atel-inf}\rightarrow\texttt{TEL-open-ATEL-NMN}\rightarrow\texttt{stem-tel-atel-nmn}

- \texttt{TEL-TEL-open-ATEL-INF}\rightarrow\texttt{stem-tel-atel-tel-inf}\rightarrow\texttt{*TEL-TEL-open-ATEL-NMN}\rightarrow\texttt{stem-tel-atel-tel-nmn}
Russian -$nie$ nominalization: derivational account

- open-INF
  - stem-inf

- TEL-open-INF
  - stem-tel-inf

- TEL-open-ATEL-INF
  - stem-tel-atel-inf

- TEL-TEL-open-ATEL-INF
  - stem-tel-atel-tel-inf
Russian nominalization: SL derivations

- *tel-nmn: the telic form cannot be nominalized;
  
  *na-ot-kry-va-nie
  stem-tel-atel-tel-nmn

- *stem-nmn: prohibit nominalization of the verbal root.
  
  *kry-nie
  stem-nmn
Russian nominalization: SL derivations

- **tel-nmn**: the telic form cannot be nominalized;
  - *na-ot-kry-va-nie*
    - stem-tel-atel-tel-nmn

- **stem-nmn**: prohibit nominalization of the verbal root.
  - *kry-nie*
    - stem-nmn
Derived vs derivational representations

Derived strings: 5-SL

\[ \text{ok} \quad \text{ot-kry-va-nie} \]
\[
\begin{align*}
\text{TEL-open} & \quad \text{ATEL-NMN} \\
\text{stem-tel-atel-nmn} &
\end{align*}
\]

Derivational strings: 2-SL

\[ \text{*na-ot-kry-va-nie} \]
\[
\begin{align*}
\text{TEL-open} & \quad \text{ATEL-TEL-NMN} \\
\text{stem-tel-atel-tel-nmn} &
\end{align*}
\]

The same difference: memorizing the illicit string vs. capturing the generalization.
Derived vs derivational representations

Derived strings: 5-SL

\textit{ot-kry-va-nie}

TEL-open-ATEL-NMN

stem-tel-atel-nmn

\textit{na-ot-kry-va-nie}

TEL-open-ATEL-TEL-NMN

stem-tel-atel-tel-nmn

Derivational strings: 2-SL

\textit{ot-kry-va-nie}

stem - tel - atel - nmn

\textit{na-ot-kry-va-nie}

stem - tel - atel - tel - nmn

The same difference: memorizing the illicit string vs. capturing the generalization.
Derived vs derivational representations

Derived strings: 5-SL

\(^{ok} \) ot-kry-va-nie
TEL-open-ATEL-NMN
stem-tel-atel-nmn

na-ot-kry-va-nie

Derivational strings: 2-SL

*na-ot-kry-va-nie
TEL-open-ATEL-TEL-NMN
stem-tel-atel-tel-nmn

The same difference: memorizing the illicit string vs. capturing the generalization.
Derived vs derivational representations

\[ \text{Derived vs. Derivational} \quad \text{Subregular Morphology} \quad \text{Russian Nominalization} \quad \text{SL Derivations} \quad \text{Conclusion} \]

\[ \text{Derived strings: 5-SL} \quad \text{Derivational strings: 2-SL} \]

The same difference: memorizing the illicit string vs. capturing the generalization.
Derived vs derivational representations

Derived strings: 5-SL

ok **ot-kry-va-nie**
TEL-open-ATEL-NMN
stem-tel-atel-nmn

na-ot-kry-va-nie

Derivational strings: 2-SL

*na-ot-kry-va-nie
TEL-open-ATEL-TEL-NMN
stem-tel-atel-tel-nmn

```
ok
T
S

na
T
S

The same difference: memorizing the illicit string vs. capturing the generalization.
```
Derived vs derivational representations

\[ \text{Derived strings: 5-SL} \]

\[ \text{o}^k \text{ot-kry-va-nie} \]
\[ \text{TEL-open-ATEL-NMN} \]
\[ \text{stem-tel-atel-nmn} \]

\[ \text{na-ot-kry-va-nie} \]
\[ \text{TEL-open-ATEL-TEL-NMN} \]
\[ \text{stem-tel-atel-tel-nmn} \]

\[ \text{ Derivational strings: 2-SL} \]

\[ \text{stem - tel - atel - nmn} \]
\[ \text{stem - tel - atel - tel - nmn} \]

The same difference: memorizing the illicit string vs. capturing the generalization.
Conclusion

Formally grounded approaches clarify ongoing linguistic debates!

Simplicity of derivational representations in morphology.

From a linguistics perspective:
- derivational representations highlight generalizations;

From a computational perspective:
- more succinct descriptions:
  - Russian aspectual sequences: 4-SL vs 2-SL;
  - Russian nominalization: 5-SL vs 2-SL;
- cross-domain complexity generalizations.
Formally grounded approaches clarify ongoing linguistic debates!

Simplicity of derivational representations in morphology.

From a linguistics perspective:
- derivational representations highlight generalizations;

From a computational perspective:
- more succinct descriptions:
  - Russian aspectual sequences: 4-SL vs 2-SL;
  - Russian nominalization: 5-SL vs 2-SL;
- cross-domain complexity generalizations.
Formally grounded approaches clarify ongoing linguistic debates!

Simplicity of derivational representations in morphology.

**From a linguistics perspective:**
- derivational representations highlight generalizations;

**From a computational perspective:**
- more succinct descriptions:
  - Russian aspectual sequences: 4-SL vs 2-SL;
  - Russian nominalization: 5-SL vs 2-SL;
- cross-domain complexity generalizations.
Conclusion

Formally grounded approaches clarify ongoing linguistic debates!

Simplicity of derivational representations in morphology.

From a linguistics perspective:
- derivational representations highlight generalizations;

From a computational perspective:
- more succinct descriptions:
  - Russian aspectual sequences: 4-SL vs 2-SL;
  - Russian nominalization: 5-SL vs 2-SL;
- cross-domain complexity generalizations.
Future work

- extend coverage of empirical phenomena;
- design a learning algorithm;
- explore parallels to subregular syntax (Graf & Heinz 2015).

The more complex the mind, the greater the need for the simplicity of play.

James T. Kirk
Future work

- extend coverage of empirical phenomena;
- design a learning algorithm;
- explore parallels to subregular syntax (Graf & Heinz 2015).

The more complex the mind, the greater the need for the simplicity of play.

*James T. Kirk*
Aksënova, Alëna, Thomas Graf and Sedigheh Moradi (2016)  
Morphotactics as Tier-Based Strictly Local Dependencies.  
In Proceedings of SIGMorPhon 2016.

Finite state morphology.  
Stanford, CA: CSLI Publications.

Kaplan, Ronald M. and Martin Kay (1994)  
Regular Models of Phonological Rule Systems.  

Graf, Thomas and J. Heinz (2015)  
Commonality in Disparity: The Computational View of Syntax and Phonology.  
Slides of a talk given at GLOW 2015, April 18, Paris, France.

Heinz, Jeffrey and W. Idsardi (2013)  
What Complexity Differences Reveal About Domains in Language.  

A fundamental misconception of modern linguistics  
Acta Linguistica Hafniensia, 35, 39?64.

McNaughton Robert and S. Papert (1971)  
Counter-Free Automata.  
On Languages Piecewise Testable in the Strict Sense

Shieber, Stuart M. 1985
Evidence against the context-freeness of natural language.
Linguistics and Philosophy. 8:333–345.

Vikner, Carl and S. Vikner (2008)
Hierarchical Morphological Structure and Ambiguity
In  L’énonciation dans tous ses états: Mélanges offerts à Henning Nølke. ed. Merete Birkelund; Maj-Britt Mosegaard Hansen; Coco Norén. Peter Lang, p. 541-560.
Non-SL morphotactics: circumfixation

- English *un-...-able* are prefix and suffix that *can* co-occur
- However, two parts of a *circumfix* *cannot* occur independently

Example (Indonesian circumfixation, Sneddon (1996))

- Circumfix *ke-...-an*, “abstract nominalizer”
  - Surrounds the stem:
    - tinggi ‘high’ → *ke-tinggi-an* ‘altitude’
  - ... or multiple stems:
    - maha-siswa ‘big-pupil’ → *ke-maha-siswa-an* ‘student affairs’
  - Parts of this affix cannot occur independently:
    - *ke-tinggi, *maha-siswa-an*

- This pattern is **not SL**: the relations between *ke-* and *-an* are **not local**.
Non-SL morphotactics: circumfixation

- English `un-...-able` are prefix and suffix that *can* co-occur
- However, two parts of a circumfix *cannot* occur independently

Example (Indonesian circumfixation, Sneddon (1996))

- Circumfix `ke-...-an`, "abstract nominalizer"
  - Surrounds the stem:
    - tinggi ‘high’ → `ke-tinggi-an` ‘altitude’
  - ... or multiple stems:
    - maha-siswa ‘big-pupil’ → `ke-maha-siswa-an` ‘student affairs’
  - Parts of this affix cannot occur independently:
    - *`ke-tinggi`, *`maha-siswa-an`

- This pattern is **not** SL: the relations between `ke-` and `-an` are *not* local.
Non-SL morphotactics: circumfixation

- English *un-...-able* are prefix and suffix that *can* co-occur
- However, two parts of a *circumfix* *cannot* occur independently

Example (Indonesian circumfixation, Sneddon (1996))

- Circumfix *ke-...-an*, “abstract nominalizer”
  - Surrounds the stem:
    - tinggi ‘high’ → *ke-tinggi-an* ‘altitude’
  - ... or multiple stems:
    - maha-siswa ‘big-pupil’ → *ke-maha-siswa-an* ‘student affairs’
  - Parts of this affix cannot occur independently:
    - *ke-tinggi*, *maha-siswa-an*

- This pattern is **not** *SL*: the relations between *ke-* and *-an* are **not** local.
Tier-based strictly local (TSL) grammars capture long-distance dependencies locally by projecting relevant items on a tier.

\[ G_{TSL} = < \]

\[ T \subseteq \Sigma \] # set of items that are projected on a tier

\[ R \] # set of k-local strings that are blocked over the tier

\[ > \]

Example (Indonesian circumfixation)

- Elements of the circumfix are projected on a tier.
- \( G = \langle \{ \text{ke, an} \}, \{ \ast \text{an-ke, ke} \times, \ast \times \text{an, an-an, ke-ke} \} \rangle \)
- Intervening stems are ignored, therefore the two parts of the circumfix are local over the tier.
Tier-based strictly local (TSL) grammars capture long-distance dependencies locally by projecting relevant items on a tier.

$$G_{TSL} = <\begin{align*} T &\subseteq \Sigma & \# \text{ set of items that are projected on a tier} \\ R &\quad & \# \text{ set of } k\text{-local strings that are blocked over the tier} \end{align*}>$$

Example (Indonesian circumfixation)

- Elements of the circumfix are projected on a tier.
- $$G = \langle \{ke, an\}, \{*an-ke, *ke\times, \times an, *an-an, *ke-ke\} \rangle$$
- Intervening stems are ignored, therefore the two parts of the circumfix are local over the tier.
Tier-based strictly local (TSL) grammars capture long-distance dependencies locally by projecting relevant items on a tier.

\[ G_{TSL} = < \]
\[ T \subseteq \Sigma \quad \# \text{ set of items that are projected on a tier} \]
\[ R \quad \# \text{ set of } k\text{-local strings that are blocked over the tier} \]
\[ > \]

Example (Indonesian circumfixation)

- Elements of the circumfix are projected on a tier.
- \( G = \langle \{ ke, an \}, \{ *an-ke, *ke\times, *\times an, *an-an, *ke-ke \} \rangle \)
- Intervening stems are ignored, therefore the two parts of the circumfix are local over the tier.
\[ G = \langle \{ \text{ke}, \text{an} \}, \{ \ast \text{an-ke}, \ast \text{ke}\ast, \ast \ast \text{an}, \ast \text{an-an}, \ast \text{ke-ke} \} \rangle \]
\[ G = \langle \{ \text{ke, an} \}, \{ \star \text{an-ke}, \star \text{ke} \times, \star \text{an, } \star \text{an-an, } \star \text{ke-ke} \} \rangle \]

\[ \text{ok} \quad \text{maha-siswa:} \]

\[ \times \quad \times \quad \text{tier of circumfix} \]

\[ \times \quad \text{maha siswa} \quad \times \]

\[ \text{ok} \quad \text{ke-maha-siswa-an:} \]

\[ \times \quad \text{ke} \quad \text{an} \quad \times \quad \text{tier of circumfix} \]

\[ \times \quad \text{ke} \quad \text{maha siswa} \quad \text{an} \quad \times \]
$G = \langle \{ \text{ke, an} \}, \{ \text{*an-ke, *ke}\ltimes, \text{*\ltimes an, *an-an, *ke-ke} \} \rangle$

$^{ok}$ maha-siswa:

\[
\begin{array}{c|c|c}
\text{\ltimes} & \text{\ltimes} & \text{tier of circumfix} \\
\hline
\text{\ltimes maha siswa} & \text{\ltimes}
\end{array}
\]

$^{ok}$ ke-maha-siswa-an:

\[
\begin{array}{c|c|c|c}
\text{\ltimes ke} & \text{an} & \text{\ltimes} & \text{tier of circumfix} \\
\hline
\text{\ltimes ke maha siswa} & \text{an}
\end{array}
\]
\[ G = \langle \{ \text{ke, an} \}, \{ *\text{an-ke, *ke} \times, *\times \text{an, *an-an, *ke-ke} \} \rangle \]

*ke-maha-siswa:
\[
\times \text{ke} \quad \times \text{ke maha siswa} \quad \times
\]

*maha-siswa-an:
\[
\times \text{maha siswa} \quad \times \text{an} \quad \times
\]
\[ G = \langle \{ \text{ke, an} \}, \{ \text{*an-ke, *ke} \times, \times \text{an}, \times \text{an-an, *ke-ke} \} \rangle \]

**ke-maha-siswa:**

\[
\times \begin{array}{c}
\times \text{ke} \\
\times \text{ke maha siswa} \\
\end{array}
\times
\]

**maha-siswa-an:**

\[
\times \begin{array}{c}
\times \text{an} \\
\times \text{maha siswa an} \\
\end{array}
\times
\]
\[ G = \langle \{\text{ke, an}\}, \{\ast\text{an-ke}, \ast\text{ke}, \ast\text{an}, \ast\text{an-an}, \ast\text{ke-ke}\} \rangle \]

**ke-maha-siswa:**

\[
\times \hspace{1cm} \text{ke} \hspace{1cm} \times \\
\times \hspace{1cm} \text{ke maha siswa} \hspace{1cm} \times
\]

**maha-siswa-an:**

\[
\times \hspace{1cm} \text{an} \hspace{1cm} \times \\
\times \hspace{1cm} \text{maha siswa an} \hspace{1cm} \times
\]